



(19)  
Bundesrepublik Deutschland  
Deutsches Patent- und Markenamt

(10) DE 103 46 623 A1 2004.10.14

(12)

## Offenlegungsschrift

(21) Aktenzeichen: 103 46 623.1

(51) Int Cl.<sup>7</sup>: B60R 21/01

(22) Anmeldetag: 08.10.2003

B60R 21/32

(43) Offenlegungstag: 14.10.2004

(66) Innere Priorität:

103 14 676.8 01.04.2003

(72) Erfinder:

Schou, Frank, 70565 Stuttgart, DE; Ringger, Klaus,  
72622 Nürtingen, DE

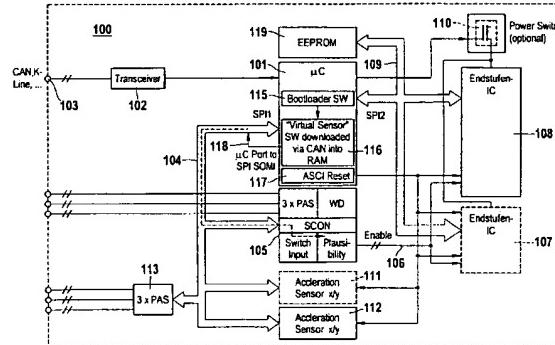
(71) Anmelder:

Robert Bosch GmbH, 70469 Stuttgart, DE

Die folgenden Angaben sind den vom Anmelder eingereichten Unterlagen entnommen

(54) Bezeichnung: Steuergerät für ein Rückhaltesystem

(57) Zusammenfassung: Es wird ein Steuergerät für ein Rückhaltesystem vorgeschlagen, das alle angeschlossenen pyrotechnischen Zündelemente zündet. Das Steuergerät erhält dafür über eine Diagnoseschnittstelle (103) ein Software-Element, das alle Zündkreise (107, 108) und den Auslösealgorithmus zur Zündung aller Zündkreise (107, 108) konfiguriert und für einen Sicherheitsbaustein derartige Sensorwerte emuliert, sodass der Sicherheitsbaustein (105) alle Zündkreise (107, 108) freigibt.



**Beschreibung**

**[0001]** Die Erfindung geht aus von einem Steuergerät für ein Rückhaltesystem nach der Gattung des unabhängigen Patentanspruchs.

**Stand der Technik**

**[0002]** In bestimmten Märkten wie Japan ist in Steuergeräten für Rückhaltesysteme eine sogenannte Funktion zur Entsorgungszündung gesetzlich vorgeschrieben. Sie dient dazu, bei der Verschrottung des Fahrzeugs alle pyrotechnischen Zünd- und Gaserzeugungselemente von Airbags und Gurtstraffern gefahrlos zu zünden, bzw. abzubrennen. Danach ist der weitere Verschrottungsvorgang ohne Gefährdung durch die Airbags und ohne Gefährdung der Umwelt möglich. Außerdem müssen die abgebrannten pyrotechnischen Elemente nicht nur aufwändig entsorgt werden, sondern können der Altmetallverwertung zugeführt werden.

**Aufgabenstellung**

**[0003]** Das erfindungsgemäße Steuergerät mit den Merkmalen des unabhängigen Patentanspruchs hat demgegenüber den Vorteil, dass ein Software-Element über eine bereits vorhandene Schnittstelle, vorzugsweise eine Diagnoseschnittstelle, des Steuergeräts in das Steuergerät eingespeist wird, wobei das Softwareelement dann alle Zündkreise und den Auslösealgorithmus zur Zündung aller Zündkreise konfiguriert und für einen Sicherheitsbaustein, der unabhängig von einem Prozessor im Steuergerät die Sensorwerte überprüft und dann gegebenenfalls in Abhängigkeit von der Überprüfung die Zündkreise freigibt, solche Sensorwerte emuliert, sodass der Sicherheitsbaustein diese Zündkreise alle freigibt. Es wird also ein Auslösefall für alle Zündkreise vorgespiegelt. Dies ermöglicht dann, dass über dieses Software-Element in einfacher Weise ohne zusätzliche Stecker über die bereits vorhandene Hardware alle Zündelemente gezündet werden können, sodass die Entsorgungszündung damit in einfachster Weise gewährleistet wird. Die Emulation der Sensorwerte ist notwendig, da auf den Sicherheitspfad über die Diagnoseschnittstelle nicht zugegriffen werden kann. Das Software-Element kann auch so ausgeführt sein, dass es lediglich ein Befehl ist, der dazu führt, dass eine weitere Software im Steuergerät aktiviert wird, die den Auslösealgorithmus und die Zündkreise zur Zündung aller vorhanden Zündelemente konfiguriert und die Sensorwerte emuliert, um letztlich die Zündelemente alle zu zünden.

**[0004]** Durch die in den abhängigen Ansprüchen angegebenen Merkmale und Weiterbildungen sind vorteilhafte Verbesserungen des im unabhängigen Patentanspruch angegebenen Steuergeräts für ein Rückhaltesystem möglich.

**[0005]** Aus Vereinheitlichungsgründen ist es besonders vorteilhaft, dass die Diagnoseschnittstelle entweder ein CAN-Bus ist oder eine K-Line ist. Die K-Line ("Kommunikations-Leitung") ist eine standardisierte Hardwareschnittstelle, über die z.B. die Werks- oder auch andere Diagnosen (Werkstattdiagnose) gemacht werden können. Vorteilhafterweise sind der Prozessor, der Sicherheitsbaustein und wenigstens ein Sensorbaustein und/oder wenigstens ein Schnittstellenbaustein zum Anschluss von externen Sensoren über einen Bus verbunden, wobei der Prozessor auf dem Bus die Sensorwerte emuliert, sodass der Sicherheitsbaustein diese emulierten Sensorwerte überprüft. Als Bus wird vorteilhafterweise der sogenannte SPI (Serial Peripheral Interface)-Bus verwendet. Die SPI-Leitung weist selbst fünf einzelne Leitungen auf. Da es sich um eine Synchronübertragung handelt, ist eine Takteleitung, die mit CLK gekennzeichnet ist, vorhanden. Für die Datenübertragung vom Master, das ist hier der Prozessor, zu einem Slave, beispielsweise einem Sensor-IC oder einem Schnittstellenbaustein, ist die MOSI (Master Out Slave In)-Leitung vorhanden. Für die Datenübertragung von einem Slave zu dem Master ist hingegen die MISO (Master In Slave Out)-Leitung vorhanden. Um den entsprechenden Slave auszuwählen, wird die CS-Leitung (Chip Select) verwendet. Um die SPI-Datenübertragung freizugeben, wird eine Enable-Leitung, die mit EN gekennzeichnet ist, verwendet. Die SPI-Leitung geht von einem Master aus und verzweigt sich dann zu den einzelnen Slaves, wobei die SPI-Leitung aber immer die fünf einzelnen Leitungen aufweist. Hier ist nun vorgesehen, dass der Prozessor über die MISO-Leitung, über die der Prozessor normalerweise Daten empfängt, diese emulierten Sensorwerte überträgt, um damit dem Sicherheitsbaustein die emulierten Sensorwerte vorzuspiegeln. Dafür ist die MISO-Leitung mit einem Input-Output-Port des Prozessors zur Übertragung der emulierten Sensorwerte verbunden.

**[0006]** Weiterhin ist im Prozessor ein Boot-Loader vorteilhafterweise vorgesehen, der das Software-Element lädt und dann sofort startet. Vorteilhafterweise kann weiterhin ein Reset-Schalter vorgesehen sein, der zum Neustart des wenigstens einen Sensorbausteins und Sicherheitsbausteins vorgesehen ist. Auch die Zündkreise können damit neu gestartet werden.

**Ausführungsbeispiel**

**[0007]** Ausführungsbeispiele der Erfindung sind in der Zeichnung dargestellt und werden in der nachfolgenden Beschreibung näher erläutert.

**[0008]** Es zeigen Fig. 1 ein Blockschaltbild des erfindungsgemäßen Steuergeräts und Fig. 2 ein Flussdiagramm.

- | Beschreibung  |   |
|---|---|
| [0009] Es wird ein Entsorgungszündungskonzept über bereits im Airbagsteuergerät vorhandene Kundiagnoseschnittstellen wie CAN oder K-Line ohne zusätzliche Schnittstellen vorgeschlagen. Es wird dabei auf die vorhandene mechanische und elektrische Hardware zurückgegriffen. Insbesondere ist bei dem erfindungsgemäßen Steuergerät vorgesehen, dass ein unabhängiger Sicherheitspfad, der durch einen Sicherheitsbaustein realisiert wird, durch die Emulation von solchen Sensorwerten getäuscht wird, die einen Auslösefall für alle Zündelemente anzeigen.  | Steuergeräts möglich ist. Dazu gehört, die Watch-Dogs zu bedienen und eine Buskommunikation zu ermöglichen. Weiterhin ist es die Aufgabe, den Airbagalgorithmus im Mikrocontroller so zu manipulieren, also zu konfigurieren, dass der Mikrocontroller die Auslösung aller Zündkreise freigibt und zündet. Eine weitere Aufgabe ist, den Sicherheitsbaustein dazu zu veranlassen, dass dieser die Plausibilität für die Endstufen-ICs liefert.  |
| [0010] Der Prozessor, der ein Mikrocontroller sein kann, wertet insbesondere über einen SPI-Bus verfügbare Sensorkanäle wie Beschleunigungswerte und Drehratenwerte aus und verarbeitet diese entsprechend den implementierten Algorithmen. Das sind solche Algorithmen, die zur Ansteuerung der Rückhaltemittel dienen, die über die Zündelemente gezündet werden, sofern es sich um pyrotechnisch auslösbarer Rückhaltemittel wie Airbag oder Gurtstraffer handelt. Der Mikrocontroller greift über einen unter Umständen zweiten SPI-Bus auf Endstufen-ICs, also die Zündkreise, zu. Die Endstufen-ICs überwachen das Zündelement und sorgen im Auslösefall dafür, dass die Auslöseenergie den angeschlossenen Zündelementen zugeführt wird.   | [0014] Ein weiterer Bestandteil eines solchen Konzepts kann optional eine einfache Schaltung zum Reset der Sensor- und Endstufen-ICs sowie des Sicherheitsbausteins, gesteuert vom Mikrocontroller sein. Die Boot-Loader Software steuert den Resetschalter. Durch diese Schaltung lässt sich das Entsorgungszündungskonzept flexibler handhaben, da nach einem solchen Reset neu konfiguriert werden kann. Dies kann unabhängig von der Initialisierungsphase des Steuergeräts erfolgen, bei deren Ende alle ICs verriegelt werden und ein Schreibzugriff danach unmöglich ist. Das Software-Element wird nunmehr einen virtuellen Sensor emulieren, der über den SPI-Bus, an dem die physikalischen Sensoren und der Sicherheitsbaustein eingeschlossen sind, emulierte Sensordaten auf den SPI-Bus einspeist. Diese emulierten Sensordaten werden vom Sicherheitsbaustein ausgewertet und veranlassen den Sicherheitsbaustein dazu, die Endstufen-ICs freizugeben. |
| [0011] Zur Realisierung des vom Mikrocontroller unabhängigen Hardwarepfads wird ein weiter IC verwendet, der hier als Sicherheitsbaustein bezeichnet wird. Dieser Sicherheitsbaustein ist an den selben SPI-Bus wie die Sensoren und der Mikrocontroller angeschlossen. Der Sicherheitsbaustein kann nur einmal nach jedem Einschalten des Steuergeräts konfiguriert werden. Danach kann nicht mehr schreibend auf ihn zugegriffen werden. Der Sicherheitsbaustein überwacht die auf dem SPI-Bus übertragenen Sensordaten und vergleicht sie mit gespeicherten Grenzwerten. Überschreitet ein Sensor definierte Schwellen, so gibt dieser Sicherheitsbaustein unabhängig vom Mikrocontroller ein Plausibilitätsignal für die Auslösung bestimmter Zündkreise. Der Sicherheitsbaustein besteht demnach aus Schaltkreisen, die in ihrer Komplexität wesentlich einfacher sind als die eines Mikrocontrollers. | [0015] Fig. 1 zeigt in einem Blockschaltbild das erfindungsgemäße Steuergerät 100. Ein Prozessor 101, hier als Mikrocontroller ausgebildet, erhält über einen Transceiver, also einen Schnittstellenbaustein 102 über eine Diagnoseschnittstelle 103 das Software-Element, mit dem die Entsorgungszündung durchgeführt wird. Die Diagnoseschnittstelle 103 ist hier ein CAN-Bus oder eine K-Line oder andere Diagnoseschnittstellen, die dazu geeignet sind.  |
| [0012] Erfindungsgemäß wird in den Mikrocontroller über eine Diagnoseschnittstelle eine spezielle Entsorgungszündungssoftware, die hier mit Software-Element beschrieben ist, direkt in einen RAM-Speicher geladen und dort gestapelt. Dazu wird eine sogenannte Boot-Loader-Software verwendet.  | [0016] Im Mikrokontroller 101 wird mit einer Boot-Loader-Software 115 das Software-Element aus einem RAM als Speicher geladen und gestartet, sodass das Software-Element den Algorithmus zur Ansteuerung von Endstufen im Prozessor 101 konfiguriert und auch die Endstufen selbst sowie die Sensorsimulation vornimmt. Die Konfiguration ist so, dass alle Zündelemente angesteuert werden. Der Algorithmus will demnach alle Zündelemente zünden. Es ist möglich, das Software-Element mit anderen Programmen zu starten.   |
| [0013] Aufgabe der Entsorgungszündungssoftware ist es, das Steuergerät mit allen Signalen zu bedienen, sodass ein unterbrechungsfreier Betrieb des  | [0017] Der Prozessor 101 ist über einen ersten SPI-Bus 104 mit einem Sicherheitsbaustein 105 sowie zwei Sensorbausteinen 111 und 112 und einem Schnittstellenbaustein 113, an den externe Sensoren angeschlossen sind, verbunden. Über einen zweiten SPI-Bus 109 ist der Mikrocontroller 101 mit einem Speicher 119, der als EEPROM ausgebildet ist, einem ersten Endstufen-IC 108 und einem zweiten  |

Endstufen-IC **107** verbunden. Über einen Datenausgang ist der Prozessor **101** optional mit einem Leistungsschalter **110** verbunden, mit dem die Energiereserviespannung auf die Endstufen **107** und **108** durchgeschaltet wird. Weiterhin ist es möglich, dass der Prozessor **101** einen sogenannten ASIC-Reset **117** durchführen kann, bei dem die Endstufen **107**, **108** und die Sensorbausteine **111** und **112** zu einem Neustart veranlasst werden. Der SPI-Bus **104** ist über seine MISO-Leitung am Prozessor **101** mit einem Input-Output-Port verbunden, um über diese eigentliche Eingangsleitung durch den Prozessor **101** die emulierten Sensorwerte des virtuellen Sensor **116**, der durch das Software-Element gebildet wird, zu übertragen. Die Eingangsleitung wird also als Ausgangsleitung verwendet. Damit wird dann der Sicherheitsbaustein **105** mittels der emulierten Sensorwerte derartig getäuscht, dass er über die Leitung **106** die Endstufen **107** und **108** freigibt. Der Mikrocontroller **101** wird dann über die Endstufen **107** und **108** die Zündung der Zündelemente veranlassen, da der Auslösealgorithmus auf dem Mikrocontroller **101** durch das Software-Element derartig konfiguriert wurde, dass die Auslösung nun erfolgen kann. Der Sicherheitsbaustein **105** überprüft und emuliert auch die Sensorwerte der über den Schnittstellenbaustein **113** extern angeschlossenen Sensoren. Das RAM ist dem Prozessor **101** zugeordnet.

**[0018]** Fig. 2 veranschaulicht, was im Steuergerät **100** passiert. Im Verfahrensschritt **200** wird das Software-Element über die Schnittstelle **103** und den Transceiver **102** in den Prozessor **101** geladen und dann von der Boot-Loader-Software **115** gestapelt. Dies erfolgt dann in Verfahrensschritt **201**. In Verfahrensschritt **202** wird dann das gestaltete Software-Element den Algorithmus im Prozessor **101** und die Endstufen **107** und **108** derartig konfigurieren, dass die Zündung aller Zündelemente erfolgen kann. Dies ist aber nur dann möglich, wenn auch der Sicherheitsbaustein **105** die Endstufen **107** und **108** freigibt. Dazu werden dann auf dem Bus **104** durch das Software-Element über einen virtuellen Sensor **116** Sensorwerte vorgetäuscht, die eine Auslösung aller Zündelemente rechtfertigen. Daraufhin gibt dann der Sicherheitsbaustein **105** über die Leitung **106** die Endstufen **107** und **108** frei. Daraufhin kann dann in Verfahrensschritt **204** die Zündung der Zündelemente über die Endstufen **107** und **108** erfolgen. Über die Schaltung **117**, die dem Prozessor **101** zugeordnet ist, können zur Konfiguration die Endstufen-ICs **107** und **108** sowie die Sensorbausteine **111** und **112**, die XY-Sensoren beinhalten, neu gestartet werden, um die Konfiguration dieser Bausteine für die Auslösung zu vereinfachen.

#### Patentansprüche

1. Steuergerät für ein Rückhaltesystem, das derart konfiguriert ist, dass das Steuergerät **(100)** alle

angeschlossenen pyrotechnischen Zündelemente zündet, dadurch gekennzeichnet, dass das Steuergerät **(100)** über eine Schnittstelle **(103)** ein Software-Element erhält, das derart konfiguriert ist, dass in Abhängigkeit von dem Software-Element alle Zündkreise **(107, 108)** und den Auslösealgorithmus zur Zündung aller Zündkreise konfiguriert werden und für einen Sicherheitsbaustein **(105)** derartige Sensorwerte emuliert werden, sodass der Sicherheitsbaustein **(105)** alle Zündkreise **(107, 108)** freigibt.

2. Steuergerät nach Anspruch 1, dadurch gekennzeichnet, dass die Schnittstelle **(103)** ein CAN-Bus ist.

3. Steuergerät nach Anspruch 1, dadurch gekennzeichnet, dass die Schnittstelle **(103)** eine K-Line ist.

4. Steuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass ein Prozessor **(101)** im Steuergerät mit dem Sicherheitsbaustein **(105)** und wenigstens einem Sensorbaustein **(111, 112)** und/oder wenigstens einem Schnittstellenbaustein **(113)** zum Anschluss von wenigstens einem externen Sensor über einen Bus **(104)** verbunden ist, wobei der Prozessor **(101)** auf dem Bus **(104)** die Sensorwerte emuliert.

5. Steuergerät nach Anspruch 4, dadurch gekennzeichnet, dass der Bus **(104)** ein SPI (Serial Peripheral Interface)-Bus ist, wobei der Prozessor **(101)** der Master ist und dadurch konfiguriert ist, dass der Prozessor **(101)** die emulierten Sensorwerte über die MISO-Leitung überträgt.

6. Steuergerät nach Anspruch 5, dadurch gekennzeichnet, dass die MISO-Leitung mit einem IO-Port **(118)** des Prozessors **(101)** zur Übertragung der Sensorwerte verbunden ist.

7. Steuergerät nach einem der vorhergehenden Ansprüche, dadurch gekennzeichnet, dass im Prozessor **(101)** eine Boot-Loader Software **(115)** vorgesehen ist, die das Software-Element lädt und startet.

8. Steuergerät nach einem der Ansprüche 2 bis 7, dadurch gekennzeichnet, dass wenigstens ein Reset-Schalter **(117)** vorgesehen ist, der zum Neustart des wenigstens einen Sensorbausteins **(111, 112)** und des Sicherheitsbausteins **(105)** sowie der Zündkreise **(107, 108)** vorgesehen ist.

Es folgen 2 Blatt Zeichnungen

## Anhängende Zeichnungen

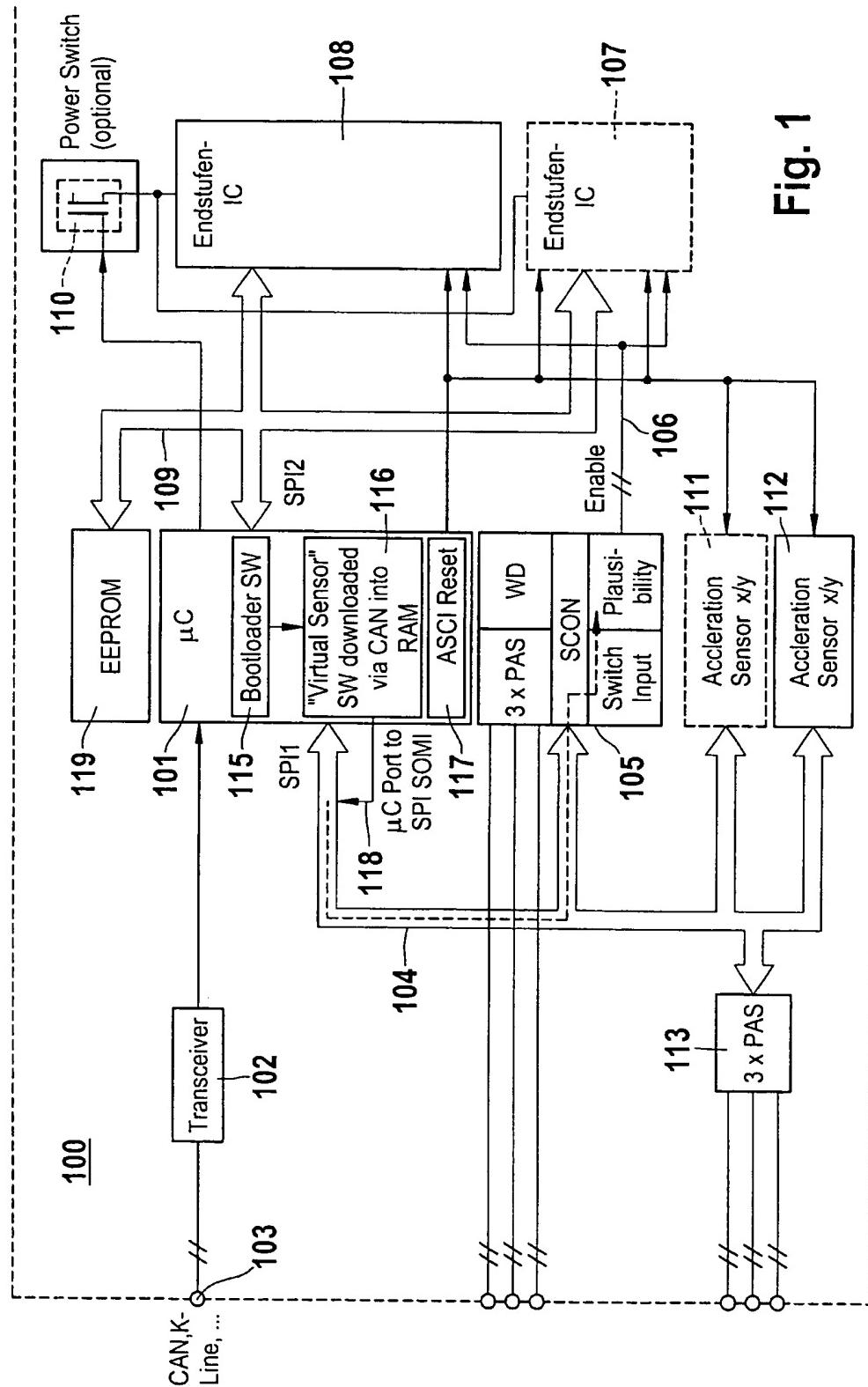
**Fig. 1**

Fig. 2

